Report on PHARO Work at Palomar Observatory
2003 Jan 30 – Feb 6

– by –

Tom Hayward

Version 1.0, 2003 Feb 27

Introduction

The purpose of this visit to Palomar by T. Hayward was to investigate a number of anomalies with PHARO, mostly related to the XPHARO software running on the Sun workstation (WS) and firmware running on the FPGA board in the instrument electronics (IE). We made a number of changes that should significantly improve the performance of the instrument. We encourage observers to monitor PHARO's performance and report any anomalies.

The author thanks the staff of Palomar Observatory for their enthusiastic support of this work. In particular, Rick Burruss assisted with much of the instrument testing and software work, Jeff Phinney maintained the dewar every day, and Dan Zieber advised on system administration issues and provided the new memory cards that were installed in the workstation. Thanks also go to Bruce Pirger at Cornell for updates to his FPGA code and to Mark Mason at EDT, Inc. for helping to diagnose the problems with the fiber optic interface code.

Fiber Optics Interface (FOI)

A major goal was to understand the cause of the various "hangs" that occasionally occur during command execution and image transfers. Our effort focused on the software and firmware related to the FOI in both the WS and IE, in consultation with the WS FOI board manufacturer EDT, Inc. and the FPGA board designer Bruce Pirger at Cornell.

The hangs occur during reads of the FOI device that normally "block" until an expected number of bytes are received from the FPGA board. On occasion, the actual number of bytes received was smaller than expected, so the read function would never return and it would have to be manually reset.

We originally believed the problem was with the FOI board because of consistent patterns in the hangs. In August 2000, we installed a new FOI board and upgraded the software driver to try to improve reliability. This did appear to reduce the frequency of the hangs, but they did not disappear completely.

During this visit, investigation of the hangs revealed a different behavior than we observed before August 2000. We concluded that occasionally the reads were simply not starting quickly enough, and the FPGA board would start sending bytes before the FOI driver was ready to receive them. The FOI buffer would overflow and data bytes would be lost. This hypothesis was confirmed by EDT's analysis of dump files which clearly showed the received number of bytes at the time of the hang was below the expected value.

To alleviate the problem, we made two main changes to the XPHARO code.

1. Multiple reads of image data were combined into one read, reducing the opportunity for race conditions.
2. A delay was added between the start of the read and the command to the FPGA board to start sending the bytes, to give the read function time to start.

In addition, we added memory to the ezra2 workstation to avoid memory swapping which could have contributed to the hangs. The `top` function now available on ezra2 is a useful way to monitor memory usage. Note that with the new management of the reads, the amount of memory used by XPHARO is larger and increases as the number of endpoint samples is increased.

After the modifications were complete, Rick Burruss and I observed no hangs of this type in several hours of testing. If hangs are still observed during operation of PHARO, one more possible improvement is to lock all the memory buffers used in the data reads. We would like to evaluate the performance of the instrument for a few observing runs before changing the code further.

One especially severe test for the FOI system is to take data with multiple cycles and the integration time set to 1.8 sec. At this rate the FPGA is almost continuously clocking the detector and sending data to the FOI asynchronously (the only pauses are the 100 msec gaps between the frames). If the EDT board is not ready to receive a frame, there is no way for it to tell the FPGA to delay or pause. The system does behave fairly reliably at this rate, but occasional failures are more likely. Increasing the integration time to 3.6 sec reduces the average workload on the workstation and may increase the reliability.

We did observe two additional FOI problems with similar behavior. One is the failure of commands just after the startup of XPHARO, which appear to be due to incomplete initialization of the FOI or FPGA board at startup. The other is the occasional appearance of "spurious" bytes in the data stream which appear to corrupt image data being transferred. We did not have time to investigate these problems fully before the end of my visit. To recover from either of type of failure, usually seen while executing a command, the following procedure is recommended:

1. Press the `STOP` button in the main XPHARO window, which stops an executing FPGA command and does a basic reset of the FOI device.
2. Click on `FOI:Send Null Cmd` to test the echoing of characters from the FPGA back to the FOI board. If that responds OK, try another normal command.

3. If there's no response to the `Null Cmd`, click on `FOI:Reset FOI`, which does a stronger reset of the FOI.
4. Again try `Send Null Cmd`.
5. If that fails, click on `Dewar:Program FPGA` to reload the FPGA code. Occasionally the automatic program at the startup of XPHARO fails.
6. If all else fails, try running `/opt/EDTsdv/sdvload` to initialize the EDT software driver. This is currently run at boot time, but may need to be rerun periodically.

Stepper Motor Homing Errors

The stepper-motor driven wheels in PHARO exhibited a problem in that when one wheel, for example the Lyot wheel, was homed, then a carousel move was attempted, the carousel would stop after only a small number of steps. We deduced that the homed wheel's magnetic position sensor was generating a spurious signal as the wheel jittered due to the vibration of the carousel motion. The FPGA firmware would interpret this as a signal from the carousel limit switches because it did not distinguish between the various sensors – a signal from any sensor would stop the mechanism currently moving.

To solve the problem, Bruce Pirger modified the FPGA code to mask off the home signals of inactive wheels. The magnetic sensor home signals are now also filtered to avoid spurious "home" readings due to voltage spikes or other transients. These changes appeared to cure the problem – Rick Burruss and I observed no failures of the carousel or any other wheel during subsequent testing.

The new home sensor control is implemented in the FPGA program file phgus218a.ttf. Observers should continue to monitor the performance of the wheels, especially when homing, and report any anomalies.

Miscellaneous Software Improvements

1. Added macro commands:
    a. `set_tint <t_int>` : Set integration time t_int in milliseconds.
    b. `set_ncycles <ncycles>`: Set number of cycles.

2. Tab characters are now permitted in macro files.

3. The TCS time is now updated properly during macro pauses. (Replaced a sleep() call in runmacro() with a more polite XtAddTimeout that recursively calls runmacro() when the pause is complete.)

4. The `take_bgd` macro command now puts data in the Background buffer.

5. The integration time counter now starts beeping at –5 seconds in the event of a failed read of the FOI.

6. Several time-variable TCS FITS keywords, such as hour angle and airmass, are now updated properly during multiple cycles.

7. Debug messages are now automatically time-stamped and written to a file pharo.debug in the current data directory. (Note that Options:Print Debug Messages toggle must be set for most messages to appear either on the screen or in the file.)